

Universidade Federal do Paraná  
Setor de Ciências Exatas  
Departamento de Estatística  
Programa de Especialização em *Data Science* e *Big Data*

Gustavo Pichorim Boiko

## **Análise comparativa de técnicas de remoção de ruído em imagens coloridas**

**Curitiba**  
**2020**

Gustavo Pichorim Boiko

**Análise comparativa de técnicas de remoção de ruído  
em imagens coloridas**

Monografia apresentada ao Programa de Especialização em Data Science e Big Data da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Luiz Eduardo Soares de Oliveira

Curitiba  
2020

## Análise comparativa de técnicas de remoção de ruído em imagens coloridas

Gustavo Pichorim Boiko<sup>1</sup>

Luiz Eduardo Soares de Oliveira<sup>2</sup>

### Resumo

A remoção de ruídos em imagens é uma das tarefas de processamento de imagens que ganhou atenção com o surgimento de métodos baseados em aprendizado de máquina. Estes métodos possibilitam a restauração de imagens afetadas por diferentes tipos de ruído, muitos destes mais complexos do que os ruídos sintéticos tradicionalmente testados. Entretanto, estes métodos em geral necessitam de treinamento extensivo com bases de dados grandes cujo preparo não é simples e, em muitos casos, inviável. Este estudo compara o comportamento e o desempenho de dois destes métodos juntamente com algumas técnicas tradicionalmente aplicadas para a tarefa de remoção de ruídos em imagens coloridas. O objetivo desta comparação é entender quão bom é o desempenho e a viabilidade desta nova abordagem quando aplicada à diferentes tipos de ruído, visando também oferecer uma plataforma que facilite a realização de novos experimentos com outras possíveis técnicas.

**Palavras-chave:** ruído em imagens, redes neurais convolucionais, processamento de imagens, imagens coloridas.

### Abstract

*Image noise removal is an image processing task that gained attention with the appearance of machine learning methods. Those methods allow restoring images containing different kinds of noise, some of them more complex than the synthetic noise traditionally tested.*

**Keywords:** *image noise, convolutional neural networks, image processing, coloured images.*

## 1 Introdução

Ruídos em imagens digitais são variações aleatórias no valor de *pixels*, causados em sua grande maioria por algum tipo de degradação durante o processo de aquisição ou de processamento, podendo ser considerado uma variável aleatória caracterizada por uma determinada densidade de probabilidade [1]. Existem diferentes tipos

de variações, sendo os mais comuns dos tipos impulsivo (com distribuição aleatória, como por exemplo o ruído sal-e-pimenta, como visto na Figura 1b, que ocorre geralmente durante a transmissão de imagens), gaussiano (variações de intensidade com distribuição gaussiana, sendo geralmente usados para modelar a degradação ocorrida durante a aquisição por câmeras fotográficas digitais, Figura 1c), poisson (seguindo uma distribuição de poisson, geralmente observados em imagens geradas por raio-x e tomografia, Figura 1d).

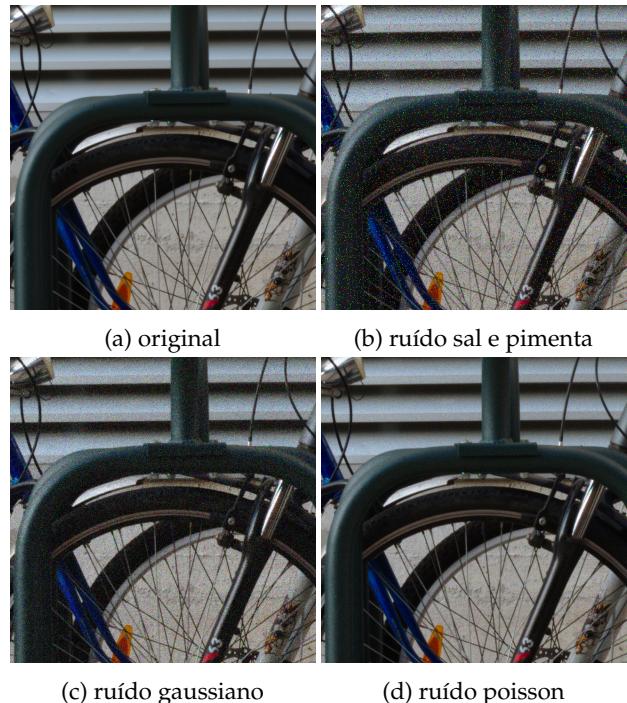


Figura 1: Exemplos de ruídos em imagens coloridas

A eliminação dos diferentes tipos de ruídos preservando o conteúdo das imagens é uma área de pesquisa bastante madura, mas nem por isso estagnada: assim como novas pesquisas e avanços tecnológicos permitem a criação de métodos mais robustos para a tarefa, a tecnologia utilizada para capturar e transmitir imagens também evolui, alterando as características do ruído encontrado nas imagens. Câmeras digitais, por exemplo, atualmente incluem não apenas um sensor de captura e uma unidade de armazenamento, mas um *pipeline* complexo de processamento de imagem. No âmbito da transmissão de imagens (sejam elas imagens estáticas

<sup>1</sup>Aluno do programa de Especialização em Data Science & Big Data, [gustavo.boiko@gmail.com](mailto:gustavo.boiko@gmail.com).

<sup>2</sup>Professor do Departamento de Informática - DINF/UFPR.

ou quadros de um vídeo), os algoritmos de compressão estão em constante evolução, também modificando os perfis de ruído introduzidos nas imagens.

Ananços nas pesquisas em redes neurais introduziram conceitos como aprendizado profundo e redes neurais convolucionais [2], cuja principal característica é a capacidade de aprender representações através dos dados, abrindo um leque de possibilidades de sua utilização em tarefas de processamento de imagens, entre elas a de remoção de ruído. Apesar de tecnicamente complexas, o princípio por trás destas técnicas é simples: dada uma topologia de rede neural pré-definida, apresentam-se amostras da imagem com ruído na camada de entrada da rede, e o resultado esperado na camada de saída. Ao repetir este procedimento com uma quantidade suficiente de amostras, a rede será capaz de capturar as características que permitem a generalização da tarefa para novas amostras ainda não apresentadas à rede.

Entretanto, devido ao seu alto custo computacional, estas linhas de pesquisa só se popularizaram com o aumento significativo no poder de processamento disponível, especialmente através da utilização de núcleos de processamento de placas gráficas (*GPUs*) para o treinamento dessas redes neurais [3].

Apesar de produzirem resultados bastante competitivos, o treinamento destas técnicas baseadas em aprendizado requer uma quantidade significativa de imagens, o que no caso de imagens com ruído não necessariamente é algo fácil de se obter. Se considerarmos, por exemplo, o ruído presente em imagens capturadas digitalmente por câmeras fotográficas, causados pela sensibilidade do sensor (ISO), garantir que as imagens obtidas com muita (ruídos) e pouca (referência) sensibilidade tenham exatamente a mesma exposição geralmente requer um pós-processamento manual cuidadoso das imagens. Uma alternativa comumente utilizada é a geração de ruído sintético nas imagens de referência, alternativa esta que, apesar de viável e mais barata, pode não representar de forma precisa o tipo de ruído que se quer eliminar.

O desempenho destes modelos normalmente está também relacionado às amostras utilizadas durante o treinamento, podendo não obter resultados satisfatórios quando aplicados a perfis de ruído diferentes dos que lhes foram apresentados.

Este trabalho apresenta uma análise comparativa de algumas técnicas tradicionais bem como técnicas de aprendizado para remoção de ruídos em imagens coloridas, a fim de demonstrar os pontos fortes e fracos de cada abordagem e identificar a viabilidade da aplicação dos mesmos em diferentes situações.

## 2 Metodologia

A fim de possibilitar uma comparação dos métodos selecionados de forma consistente e padronizada, foi desenvolvida uma ferramenta [4] de arquitetura modular que permite o uso de diferentes técnicas, métricas e conjunto de dados de forma simples, bem como a extração e armazenamento de resultados para avaliação.

Esta ferramenta é composta de quatro diferentes componentes, cujos detalhes são discutidos nas próximas seções:

- ▶ *dataset*: conjunto de dados a ser utilizado. Este pode ser simétrico (com pares de imagens referência e ruidosa), ou conter apenas as imagens de referência para uso em conjunto com ruídos sintéticos.
- ▶ *noiser*: método capaz de produzir ruído sintético em uma imagem de referência.
- ▶ *metric*: métrica numérica de comparação de uma determinada imagem em relação a uma referência.
- ▶ *denoiser*: um método capaz de processar imagens de entrada para remoção de ruído. Não há distinção entre métodos tradicionais ou baseados em aprendizado de máquina (embora estes sejam discutidos em seções distintas adiante).

### 2.1 Conjunto de dados

Para a realização dos experimentos, foi escolhido o conjunto de dados *Natural Image Noise Dataset* [5], composto de cenas capturadas com posicionamento de câmera fixo, variando a sensibilidade (ISO), mas preservando os níveis de exposição. Para cada uma das cenas, foi selecionado um par de imagens: a de menor sensibilidade, usada como imagem de referência, e a de maior sensibilidade, usada como imagem ruidosa a ser tratada. A Figura 3 mostra um desses pares.

Originalmente, o conjunto de dados possui imagens de tamanhos diferentes (Figura 2). A fim de minimizar a interferência dessa diferença e de viabilizar a realização de um número maior de experimentos, as imagens foram recortadas para as menores dimensões encontradas (1311 pixels de largura por 1403 pixels de altura).

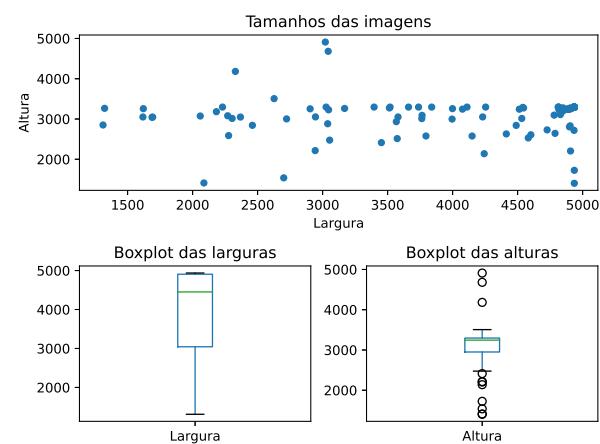


Figura 2: Tamanhos das imagens do conjunto de dados.

### 2.2 Imagens com ruído

Além da imagem ruidosa já presente no conjunto de dados, o experimento foi realizado também através da adição de ruído sintético às imagens de referência. Os seguintes tipos de ruído foram testados:

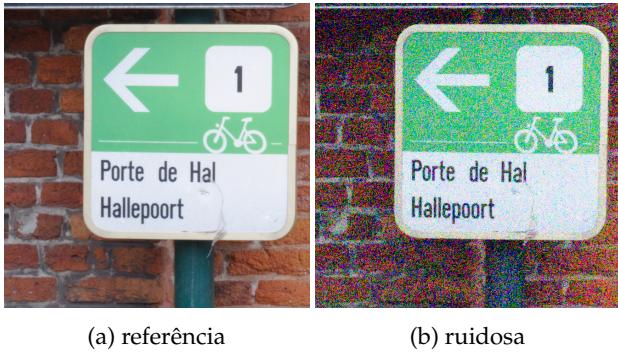


Figura 3: Recorte das imagens de referência e ruidosa de uma das cenas do conjunto de dados *Natural Image Noise Dataset* [5]

- ▶ **gaussian:** Ruído gaussiano aditivo com variância de 0,01.
- ▶ **snp:** Ruído sal e pimenta com probabilidade 0,05 e com a mesma proporção entre ruído sal e ruído pimenta.
- ▶ **poisson:** Ruído com distribuição de Poisson gerado a partir dos dados da imagem.
- ▶ **speckle:** Ruído multiplicativo aplicado da seguinte maneira:

$$\text{sada} = \text{imagem} + n * \text{imagem}$$

onde  $n$  é um ruído uniforme com média 0 e variância 0,01.

### 2.3 Métricas de comparação

A comparação entre os resultados é feita através da extração de diferentes métricas entre a imagem resultante ( $g$ ) e a imagem de referência ( $f$ ). A literatura oferece diversas métricas com comportamento e propósito diferentes. Algumas representam diferenças absolutas entre os valores de cada ponto das imagens, outras tentam representar outros aspectos, entre eles a similaridade entre o conteúdo das imagens. As seguintes métricas foram extraídas durante os experimentos:

- ▶ **MSQE:** erro médio quadrático, representa a média do quadrado das diferenças entre os pixels das imagens, sendo que quanto menor o valor resultante, melhor o resultado (Equação 1).

$$\text{MSQE} = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - g(x, y)]^2 \quad (1)$$

- ▶ **NRMSE:** raiz do erro médio quadrático normalizado, representa a proporção entre a raiz do erro médio quadrático e a raiz da média dos quadrados dos valores da imagem original, estando os valores normalizados entre 0 e 1, sendo que quanto menor o valor encontrado, melhor é o resultado (Equação 2).

$$\text{NRMSE} = \frac{\sqrt{\text{MSQE}}}{\sqrt{\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y)]^2}} \quad (2)$$

- ▶ **PSNR:** relação sinal-ruído de pico, representa a proporção entre o maior nível de sinal da imagem de referência ( $L_{max}$ ) e a média do ruído representado pelo valor do MSQE sendo que quanto maior o valor encontrado, melhor é o resultado (Equação 3).

$$\text{PSNR} = 10 \log_{10} \left( \frac{L_{max}^2}{\text{MSQE}} \right) \quad (3)$$

- ▶ **SSIM:** média do índice de similaridade estrutural [6], calculado ponto a ponto entre as imagens de resultado e referência. Quanto maior o índice obtido, maior a similaridade entre as imagens.

Devido ao custo computacional elevado da execução dos métodos, e para garantir a integridade da análise, ao final da aplicação de cada uma das técnicas à uma imagem, todas as métricas entre a imagem resultante e a de referência foram calculadas e armazenadas.

Além das métricas dos resultados, também foram calculadas as métricas entre a imagem ruidosa a ser processada e a de referência, a fim de analisar o comportamento dos métodos em relação aos diferentes níveis de ruído encontrados. Neste aspecto, o conjunto de dados utilizado contém imagens cujos níveis de ruído estão relativamente bem distribuídos entre imagens com pouco e muito ruído (Figura 4).

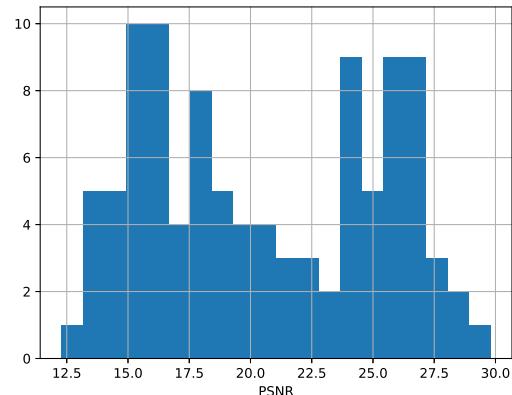


Figura 4: Histograma da relação sinal-ruído de pico (PSNR) das imagens do conjunto de dados.

### 2.4 Medição do tempo de execução

Para a medição do tempo de execução de cada método nas imagens processadas, estes foram executados de forma sequencial a fim de minimizar possíveis interferences causadas pela paralelização de tarefas. O tempo medido equivale ao tempo gasto realizando a tarefa, excluindo-se o tempo gasto lendo as imagens, calculando as métricas e armazenando os resultados.

Alguns métodos necessitam que as imagens estejam representadas usando valores em ponto flutuante, e portanto foi necessária uma conversão. O tempo gasto com essa conversão está incluso no tempo medido, mas é suficientemente pequeno para poder ser desconsiderado.

No âmbito deste experimento, o tempo total de execução não reflete com exatidão o custo computacional das técnicas, visto que as técnicas tradicionais (Seção 2.5) não fazem uso de nenhum paralelismo para o processamento de uma imagem individualmente. As técnicas baseadas em aprendizado, por sua vez, fazem uso de bibliotecas que utilizam paralelismo automaticamente para a realização das tarefas. Mesmo assim, esta métrica serve como base para uma análise da viabilidade da aplicação de cada técnica em casos de uso reais.

Para as outras variantes testadas (com ruídos sintéticos descrita na Seção 2.2), os testes foram realizados através do processamento de diversas imagens em paralelo, e portanto a medição do tempo de execução não foi analisada.

## 2.5 Métodos tradicionais de remoção de ruído

Dentre as muitas técnicas tradicionalmente usadas para a remoção de ruídos, foi selecionado um subconjunto abrangendo tanto métodos mais simplistas (filtros passa-baixa, por exemplo), quanto mais elaborados a fim de identificar seu desempenho em relação aos métodos discutidos na Seção 2.6. As seguintes técnicas de remoção de ruído foram analisadas:

- ▶ **nlmeans**: médias não locais [7].
- ▶ **bilateral**: filtro bilateral [8].
- ▶ **tvchambolle**: minimização da variação total da imagem [9].
- ▶ **wavelet**: limiar de *wavelet* adaptativo [10].
- ▶ **fastnlmeans**: variante melhorada [11] do método *nlmeans*.
- ▶ **blur**: suavização de imagens utilizando o filtro da média [12].
- ▶ **gaussianblur**: suavização de imagens utilizando um filtro gaussiano [12].
- ▶ **medianblur**: suavização de imagens utilizando o filtro da mediana [12].
- ▶ **bm3d**: transformadas de domínio para filtragem colaborativa de ruído estacionário correlacionado [13].

Os métodos acima possuem parâmetros que modificam seu comportamento e afetam a qualidade dos resultados produzidos. Com o objetivo de encontrar valores e combinações de parâmetros que produzam bons resultados, foi realizada uma busca aleatorizada com validação cruzada usando *k-fold*, utilizando 70% do conjunto de dados, recortando uma janela de cada imagem em uma posição aleatória. A tabela 1 mostra a lista completa de parâmetros utilizados nos testes.

## 2.6 Métodos baseados em aprendizado

Devido à disponibilidade de recursos, a seleção das técnicas de remoção de ruído baseadas em aprendizado priorizou aquelas que possuíam tanto uma implementação quanto modelos pré-treinados disponíveis. Boa

Método	Parâmetros
nlmeans	patch_size: 3 patch_distance: 17 h: 0.51
bilateral	win_size: 3 sigma_color: 0.1 sigma_spatial: 1 bins: 10000
tvchambolle	weight: 0.1 eps: 0.0002 n_iter_max: 200
wavelet	sigma: 0.8 wavelet: "db6" converg2ycbcr: falso
fastnlmeans	h: 14 h_color: 29 template_window_size: 3 search_window_size: 23
blur	kernel_size: 11
gaussianblur	kernel_size: 11 sigma_x: 3.9 sigma_y: 2.418
medianblur	kernel_size: 15
bm3d	sigma_psd: 30.0

Tabela 1: Valores dos parâmetros dos métodos utilizados nos testes

parte das técnicas avaliadas seriam inviáveis devido a limitações de tempo e hardware caso sua implementação ou modelos previamente treinados não estivessem disponíveis.

Foram integrados três modelos diferentes para análise:

- ▶ **cycleisp**: este método implementa uma simulação de um *pipeline* de câmera fotográfica em uma rede neural convolucional, podendo ser utilizado nas direções direta e reversa [14]. O modelo foi disponibilizado com dois conjuntos de pesos que foram obtidos ao treiná-lo com os conjuntos de dados *SIDD* [15] e *DND* [16]. Para os experimentos os pesos do conjunto de dados *DND*.
- ▶ **cbdnet**: implementa uma rede neural convolucional para remoção de ruído às cegas, através do uso da arquitetura da rede, aprendizado assimétrico e modelagem de ruído [17]. A implementação utilizada para este modelo foi também treinada utilizando o conjunto de dados *DND* [16], mas além disso foi também treinada utilizando ruído sintético e a combinação dos dois anteriores. Para a análise foram utilizados os pesos do treinamento combinado de imagens reais (*DND*) e com ruído sintético.
- ▶ **deepimageprior**: ao contrário dos métodos anteriores, este método não requer treinamento prévio para seu funcionamento. Ele baseia-se na premissa

de que apenas a estrutura de uma rede neural convolucional somada às próprias características é suficiente para a restauração da imagem [18]. Este método, apesar de estar disponível na ferramenta implementada, precisou ser descartado para os experimentos devido ao seu alto custo computacional quando executado exclusivamente em CPU. Em testes preliminares o processamento de cada imagem dura diversas horas, tornando inviável a conclusão dos experimentos.

### 3 Resultados

Foram realizados testes com cinco tipos de ruídos diferentes, e para cada um destes quatro métricas diferentes foram coletadas. Os resultados apresentados nesta seção representam apenas uma parte dos experimentos, sendo que o restante dos dados dos testes realizados (à exceção das imagens) foram disponibilizados na íntegra juntamente com a ferramenta desenvolvida [4].

Baseado nos resultados obtidos, foi calculado o coeficiente de correlação de Pearson entre os valores das métricas (Figura 5) e observou-se que existe uma correlação entre os valores obtidos, permitindo a escolha de apenas uma das métricas (PSNR) para discussão nesta seção.

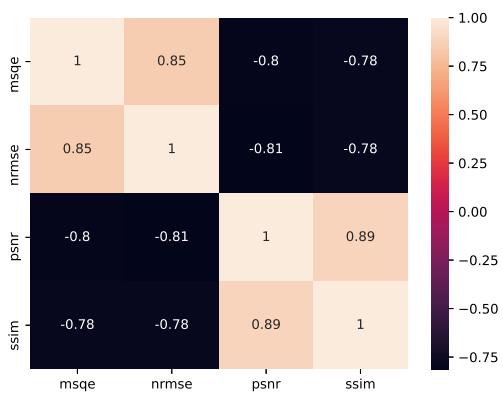


Figura 5: Correlação entre o valor das métricas nos resultados obtidos.

#### 3.1 Desempenho médio das técnicas

O desempenho médio das técnicas segundo a métrica PSNR é apresentado na Tabela 2. Como esperado, quando executados usando as imagens ruidosas originais do conjunto de dados, os métodos baseados em aprendizado conseguiram bons resultados, visto que este perfil de ruído é similar ao que foi originalmente utilizado no treinamento destes.

Entretanto, o conjunto de dados possui imagens com níveis de ruído diferentes (Figure 4), e ao compararmos o desempenho das técnicas nos extremos com maior e

menor intensidade de ruído (Tabela 3), pode-se observar que em imagens com níveis elevados de ruído seu desempenho foi semelhante ao de técnicas tradicionais simples (como o filtro da mediana).

Nos testes com ruídos sintéticos, apesar de seu desempenho ter melhorado em alguns casos, os métodos baseados em aprendizado produziram em média resultados inferiores aos de métodos tradicionais, especialmente em relação ao método BM3D [13] que se mostrou bastante robusto nos testes.

Os testes envolvendo ruído sal e pimenta foram os que levaram os métodos CycleISP e CBDNet a obter seus piores resultados de acordo com a métrica PSNR. Entretanto, uma inspeção visual mostrou que alguns métodos tradicionais com valor de PSNR maior do que os obtidos pelo método CycleISP não foram capazes de remover todos os artefatos de ruído (Figura 6c), ou causaram uma degradação do conteúdo (Figura 6d). É possível ver que, mesmo com um valor de PSNR baixo, o método CycleISP foi capaz de produzir um resultado visualmente mais agradável que alguns dos outros mencionados (Figura 6f), entretanto em muitas das imagens foi possível notar diferenças de cor e intensidade no resultado (o que possivelmente afetou negativamente seu PSNR).

Conforme mencionado anteriormente, neste teste o método BM3d obteve bons resultados (Figura 6e), tanto pela métrica PSNR quanto na inspeção visual, restando apenas poucos pontos ruidosos na imagem.

#### 3.2 Tempo de execução

O tempo médio de execução das técnicas, observado na Figura 7, mostra que existe uma diferença grande, mesmo entre as técnicas tradicionais, no tempo gasto para a realização da tarefa. A maioria dos métodos tradicionais, no entanto, precisou de menos tempo que as técnicas baseadas em aprendizado para terminar o processo.

Dois dos métodos tradicionais se mostraram bastante custosos em termos de tempo de processamento (BM3D e NLMeans), mas ao compará-los aos métodos de aprendizado deve-se levar em conta o fato de que estes fizeram uso de paralelismo para a obtenção da imagem resultado (conforme discutido na Seção 2.4).

Assim como nos métodos tradicionais, nos de aprendizado também há uma variação grande no tempo necessário para processar cada imagem. O método CBDNet, por exemplo, é consideravelmente mais rápido que o método CycleISP. Ainda assim, ambos foram viáveis de execução durante os testes, ao contrário do método Deep Image Prior [18], que em testes preliminares teve um tempo de execução medido de aproximadamente 30 horas para cada imagem processada, o que o tornou inviável de ser avaliado, tendo sido desconsiderado no contexto desta análise.

Filtro	Ruído				
	original	gaussian	poisson	snp	speckle
bilateral	23.0 ( $\pm 4.8$ )	26.9 ( $\pm 0.4$ )	32.5 ( $\pm 1.5$ )	22.5 ( $\pm 1.7$ )	31.7 ( $\pm 1.7$ )
blur	25.8 ( $\pm 3.3$ )	28.7 ( $\pm 3.3$ )	29.3 ( $\pm 3.7$ )	28.1 ( $\pm 2.9$ )	29.3 ( $\pm 3.7$ )
bm3d	25.0 ( $\pm 6.4$ )	33.2 ( $\pm 2.3$ )	34.5 ( $\pm 2.7$ )	31.5 ( $\pm 2.1$ )	34.4 ( $\pm 2.7$ )
fastnlmeans	25.4 ( $\pm 4.7$ )	28.3 ( $\pm 2.6$ )	30.9 ( $\pm 3.0$ )	27.0 ( $\pm 2.4$ )	30.8 ( $\pm 3.0$ )
gaussianblur	26.1 ( $\pm 3.5$ )	29.8 ( $\pm 2.9$ )	30.8 ( $\pm 3.6$ )	29.0 ( $\pm 2.6$ )	30.7 ( $\pm 3.5$ )
medianblur	25.9 ( $\pm 3.5$ )	28.3 ( $\pm 3.5$ )	28.9 ( $\pm 3.8$ )	29.1 ( $\pm 4.0$ )	28.8 ( $\pm 3.8$ )
nlmeans	25.4 ( $\pm 5.7$ )	31.0 ( $\pm 2.5$ )	31.7 ( $\pm 2.9$ )	25.6 ( $\pm 2.3$ )	31.7 ( $\pm 2.9$ )
tvchambolle	25.1 ( $\pm 5.0$ )	31.1 ( $\pm 1.7$ )	33.7 ( $\pm 2.8$ )	23.4 ( $\pm 1.1$ )	33.6 ( $\pm 2.6$ )
wavelet	25.1 ( $\pm 3.4$ )	28.9 ( $\pm 3.0$ )	29.7 ( $\pm 3.7$ )	28.2 ( $\pm 2.7$ )	29.7 ( $\pm 3.6$ )
cbdnet	27.8 ( $\pm 5.7$ )	23.4 ( $\pm 1.1$ )	31.7 ( $\pm 2.0$ )	19.9 ( $\pm 0.9$ )	30.4 ( $\pm 2.5$ )
cycleisp	28.1 ( $\pm 4.4$ )	26.5 ( $\pm 2.1$ )	31.5 ( $\pm 2.4$ )	23.6 ( $\pm 3.3$ )	30.3 ( $\pm 2.7$ )

Tabela 2: PSNR médio dos métodos nos testes com os diversos tipos de ruído

Filtro	Imagens		
	todas	ruidosas	limpas
bilateral	23.0 ( $\pm 4.8$ )	16.8 ( $\pm 1.0$ )	29.3 ( $\pm 1.1$ )
blur	25.8 ( $\pm 3.3$ )	22.0 ( $\pm 1.4$ )	29.0 ( $\pm 2.8$ )
bm3d	25.0 ( $\pm 6.4$ )	16.6 ( $\pm 1.1$ )	32.7 ( $\pm 1.5$ )
fastnlmeans	25.4 ( $\pm 4.7$ )	19.0 ( $\pm 2.0$ )	30.2 ( $\pm 2.0$ )
gaussianblur	26.1 ( $\pm 3.5$ )	21.7 ( $\pm 1.3$ )	30.0 ( $\pm 2.5$ )
medianblur	25.9 ( $\pm 3.5$ )	22.2 ( $\pm 2.1$ )	28.9 ( $\pm 2.9$ )
nlmeans	25.4 ( $\pm 5.7$ )	17.0 ( $\pm 1.6$ )	31.1 ( $\pm 1.9$ )
tvchambolle	25.1 ( $\pm 5.0$ )	18.4 ( $\pm 1.2$ )	31.3 ( $\pm 1.7$ )
wavelet	25.1 ( $\pm 3.4$ )	20.9 ( $\pm 1.2$ )	29.0 ( $\pm 2.5$ )
cbdnet	27.8 ( $\pm 5.7$ )	20.3 ( $\pm 2.4$ )	33.7 ( $\pm 1.6$ )
cycleisp	28.1 ( $\pm 4.4$ )	22.3 ( $\pm 3.0$ )	32.6 ( $\pm 1.2$ )

Tabela 3: Comparação do PSNR médio entre o conjunto de dados completo e os extremos mais e menos ruidosos das imagens ruidosas originais.

## 4 Conclusão

A escolha de métodos baseados em aprendizado para a remoção de ruído em imagens coloridas deve levar em consideração diversos aspectos. O primeiro deles é a disponibilidade de um conjunto de imagens de teste com ruído igual ou similar ao que se pretende remover. O desempenho destes métodos em imagens cuja característica de ruídos seja diferente do que foram treinados para remover faz com que seu desempenho seja afetado negativamente.

Em casos nos quais o tipo de ruído é bem conhecido e facilmente classificável, a aplicação de técnicas tradicionais pode ser mais adequada, especialmente se restrições de poder computacional (tempo de resposta, limitação de processador e de memória) forem um aspecto importante do problema ser solucionado.

A popularização do uso de placas gráficas (GPU), assim como o barateamento destas é um fator que pode contribuir positivamente para o uso de técnicas de aprendizado em um maior número de aplicações.

Apesar de as métricas utilizadas serem bastante robustas, o desenvolvimento de métricas que se assemelhem mais à percepção de ruído pelo sistema visual humano pode contribuir para uma melhor comparação e aprimoramento das técnicas pesquisadas.

## Agradecimentos

Gostaria de agradecer à minha esposa Elenise, pela compreensão e apoio durante o período do curso e da elaboração do trabalho. Ao orientador Prof. Luiz Eduardo pelo direcionamento e contribuições ao trabalho. À empresa SUSE Software Solutions Germany GmbH pelo apoio financeiro e pela flexibilidade fornecida para a conclusão deste curso.

## Referências

- [1] H. Pedrini and W.R. Schwartz. *Análise de Imagens Digitais: Princípios, Algoritmos, e Aplicações*. Thomson Learning, 2008.
- [2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25:1106–1114, 2012.
- [4] G.P. Boiko. Denoiser Comparator, 2020. [https://github.com/boiko/denoiser\\_comparator](https://github.com/boiko/denoiser_comparator).
- [5] B. Brummer and C. Vleeschouwer. Natural Image Noise Dataset. In *CVPR Workshops*, page 1777–1784. Computer Vision Foundation / IEEE, 2019.
- [6] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility

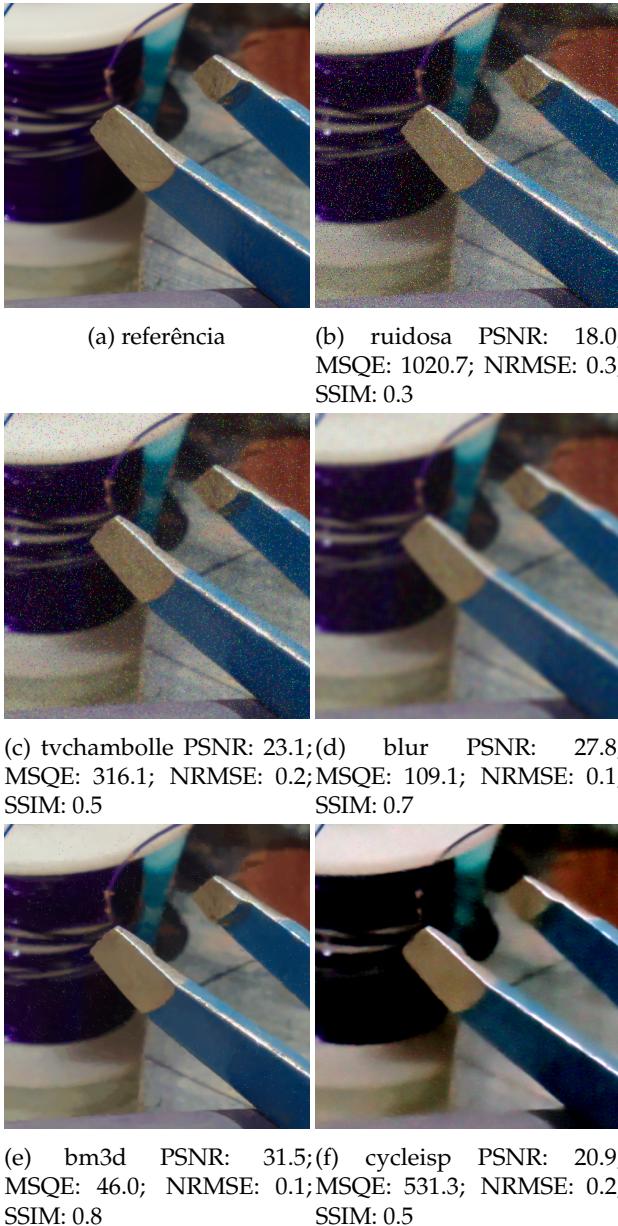


Figura 6: Recorte de um resultado do teste envolvendo ruído sal e pimenta.

- to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.
- [7] A. Buades, B. Coll, and J. M. Morel. A Non-Local Algorithm for Image Denoising. In *CVPR* (2), page 60–65. IEEE Computer Society, 2005.
  - [8] C. Tomasi and R. Manduchi. Bilateral Filtering for Gray and Color Images. In *Int. Conf. on Computer Vision '98: Proceedings of the Sixth Int. Conf. on Computer Vision*, page 839, Washington, DC, USA, 1998.
  - [9] A. Chambolle. An Algorithm for Total Variation Minimization and Applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.
  - [10] S. G. Chang, Bin Yu, and M. Vetterli. Adaptive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Processing*, 9(9):1532–1546, 2000.

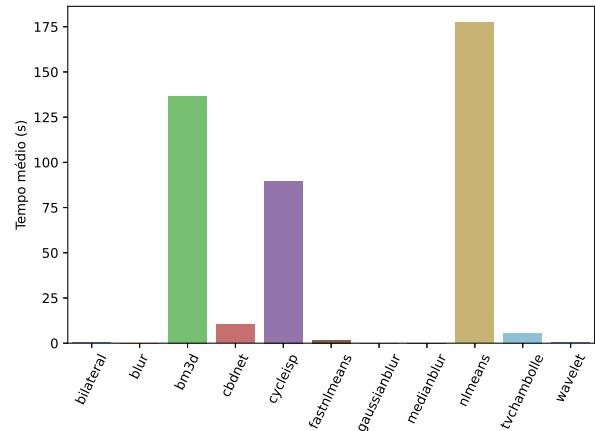


Figura 7: Tempo médio de execução de cada uma das técnicas.

- tive wavelet thresholding for image denoising and compression. *IEEE Trans. Image Processing*, 9(9):1532–1546, 2000.
- [11] A. Buades, B. Coll, and J.M. Morel. Non-Local Means Denoising. *IPOL Journal*, 1, 2011.
  - [12] R. C. Gonzalez and R. E. Woods. *Digital image processing*. Prentice Hall, Upper Saddle River, N.J., 2008.
  - [13] Y. Mäkinen, L. Azzari, and A. Foi. Exact Transform-Domain Noise Variance for Collaborative Filtering of Stationary Correlated Noise. In *ICIP*, page 185–189. IEEE, 2019.
  - [14] S.W. Zamir, A. Arora, S. Khan, M. Hayat, F.S. Khan, M.H. Yang, and L. Shao. CycleISP: Real Image Restoration via Improved Data Synthesis, March 2020.
  - [15] A. Abdelhamed, S. Lin, and M.S. Brown. A High-Quality Denoising Dataset for Smartphone Cameras. In *CVPR*, page 1692–1700. IEEE Computer Society, 2018.
  - [16] T. Plotz and S. Roth. Benchmarking Denoising Algorithms with Real Photographs. In *CVPR*, page 2750–2759. IEEE Computer Society, 2017.
  - [17] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang. Toward Convolutional Blind Denoising of Real Photographs. In *CVPR*, page 1712–1722. Computer Vision Foundation / IEEE, 2019.
  - [18] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep Image Prior. In *CVPR*, page 9446–9454. IEEE Computer Society, 2018.